Ontology quality and fitness: A survey of software support

Ontology Summit February 14, 2013

Michael Denny
msdenny@mitre.org

Survey consideration: Casting evaluation factors as capabilities

- At this juncture, many ontology evaluation factors/ dimensions/criteria identified in the tracks are yet to be fully developed and remain general, vague, or abstract. For instance, "elegance", "parsimony", "reliability", and "consistency".

- They could easily translate into any number of different ontology software capabilities and metrics. In a survey, this under-specification could lead to software providers being unable to differentiate their software capabilities from those of other software.

- Consequently, the subject items composing this survey are representative capabilities rather than the evaluation factors addressed by these capabilities.

Survey consideration: Limiting factors to those amenable to software support

- In some cases, the evaluation factors identified in a track reflect aspects of an ontology that are not so directly related to software capabilities.

- For example, the cost of development or the clarity of term definitions may be mostly a human determination.

- Ontology quality and fitness factors covered by capabilities included in the survey list are only those likely to be amenable to assessment or control by software.

Survey consideration: Putting it in the perspective of users & builders

- When characterizing the evaluation and support of ontology quality and fitness offered in software, the most direct perspective is that of the ontology user or builder.

- The approach covers both ontology assessment tools and ontology development tools.

- The distinctions made in other tracks about "intrinsic – extrinsic" and "implicit – explicit" dichotomies were not carried over to the survey list structure. These distinctions may be hard to determine in respect to the role of software.

Survey consideration: Organizing software capabilities by ontology development phase

- Potential software capabilities were compiled as those that may address the quality or fitness (evaluation factors) of ontologies identified in the tracks. The compilation was arranged into general phases of ontology development to make it a bit more manageable.

- Exploration Phase
  Management Phase
  Design Phase
  Build Phase
  Validation Phase
  Integration & Use Phase
  Maintenance Phase

Survey consideration: Making it palatable to respondents

- The resulting survey is rather long and, unless respondents are able to restrict their focus to only a few development phases, the identification and explanation of their software's relevant capabilities may take some time.

- One possibility to ease the burden is to waive the requirement for explaining each relevant software capability and treating the survey as a simple checklist.

Instructions for
survey of software support

- Software capabilities to evaluate or promote the quality or fitness of an ontology are arranged below into several general phases of ontology development.

- Identify which of these capabilities is delivered by your software environment or tool, and explain how the software achieves this particular capability.

- If a capability of your software supports ontology quality or fitness but is not included in the list, please add it to the appropriate phase.

Exploration Phase Support

1. Find ontologies with specific domain coverage
2. Compare domain coverage across ontologies
3. Assess or characterize breadth of domain coverage of an ontology
4. Assess or characterize depth/detail of domain coverage of an ontology
5. Assess or characterize use considerations of ontology including licensing, training, cost, updating, software requirements, and security
6. Assess complexity of concept definitions (necessary & sufficient properties) and use of axioms
7. Identify number and ratio of concepts, relations, and subclassing
8. Identify frequency of use of ontology language features in an ontology
9. Identify age and use statistics of an ontology and its versions
10. Assess the inferencing power of an ontology
11. Rate ontologies on their popularity or review feedback
12. Other ontology quality and fitness support for exploration phase (describe capability)

Management Phase Support

1. Distinguish between conceptual and operational ontologies and objectives
2. Generate ontology application and performance requirements
3. Organize and track the life cycle requirements and progression of an ontology
4. Organize and maintain a collection of related ontologies and ontology modules
5. Provide statistics on ontology versioning and use
6. Guide ontology development per a quality assurance methodology
7. Assess the reusability of a planned ontology
8. Other ontology quality and fitness support for management phase (describe capability)

Design Phase Support

1. Guide formulation of ontology requirements
   a. Domain scope and detail requirements
   b. Goodness of design requirements
   c. Data and information requirements
   d. Semantic and reasoning requirements
   e. Interface requirements
   f. Level of effort requirements
2. Facilitate framing a set of competency questions exemplifying ontology objectives
3. Choose metaphysical methodology
4. Choose extant ontologies for reuse
   a. Choose top ontology
   b. Choose core ontologies
5. Apply a style of ontological analysis to design
6. Guide design to optimize normalization, factoring, and simplicity of ontology
7. Guide application of open world or closed world semantics
8. Guide ontology design to achieve inferencing requirements
9. Guide ontology design to achieve scalability requirements
10. Enable adjustable query and inference performance
11. Offer design patterns based on current ontology or design context
    a. Application patterns
    b. Ontology language patterns
    c. Methodology patterns
    d. Other patterns
12. Integrate selected design patterns into an ontology design
13. Assess correctness or performance of an ontology design
14. Guide and facilitate modularization of ontology
15. Guide ontology design for visualization
16. Other ontology quality and fitness support for design phase (describe capability)

Build Phase Support: General

1. Choose ontology language
   a. Formal logic language
   b. RDF support
2. Mix ontology languages
3. Generate ontology code from requirements specification
4. Generate or enforce selected design patterns in ontology code
5. Feedback on consistency of granularity and regularity of domain ontology structure
6. Manage lexical naming and annotation of ontology elements
7. Ensure proper use and coding of RDF and Web resources
8. Enforce proper use and coding of URIs
9. Produce the current terminological inferences of an ontology
10. Optimize query and inference performance
11. Detect violations of domain, referential, or semantic integrity
12. Overall, detect and correct coding errors or inconsistencies
13. Other ontology quality and fitness support for build phase (describe capability)

Build Phase Support: Concepts

1. Ensure proper separation and coding of concepts and facts
2. Ensure "kind-of" nature and consistency of subclassing
3. Guide subclass versus class individual determinations
4. Monitor depth of subclassing consistency across a domain ontology
5. Generate prototypical instances (individuals) to help verify class intent and class subsumption
6. Detect and guide use of multiple inheritance
7. Guide use of disjointness axioms
8. Guide use of necessary & sufficient properties in concept definitions
9. Guide use of existential versus universal quantification in class restrictions
10. Enforce proper use of conjunctions versus disjunctions

Build Phase Support: Relations

1. Ensure proper use and coding of relations
2. Ensure proper use and coding of data types
3. Assess/enforce consistency and completeness of range and domain constraints
4. Assess/enforce consistency and completeness of inverse relations
5. Guide use of property characteristics (e.g., reflexivity) and negation
6. Guide use of disjointness axioms
7. Detect and guide use of multiple inheritance
8. Ensure proper use and coding of transitive relations

Validation Phase Support

1. Verify that ontology requirements are met
2. Assess query performance
   a. Query precision and recall
   b. Query time
3. Assess inference performance
   a. Inferencing time
   b. Inferencing completeness
4. Generate characteristic queries and tests
5. Accept validation test sets or inputs
6. Assure computability of ontology
   a. Semantically adequate
   b. Mathematically complete
7. Verify that two ontologies are interoperable
8. Validate the intended functionality of software using the ontology
9. Validate instance data conforming to an ontology
10. Assess accuracy, correctness, and completeness of ontology terminological content
11. Guide or adjust ontology in accord with validation results
12. Other ontology quality and fitness support for validation phase (describe capability)

Integration & Use Phase Support

1. Export ontology in different languages
2. Integrate ontology with other ontologies
3. Integrate ontology with other information system resources
4. Assess or track user experience with ontology
5. Enable user to modify or extend ontology to address deficiencies
6. Create mappings from ontologies to/from data and data sources
7. Other ontology quality and fitness support for integration and use phase (describe capability)

Maintenance Phase Support

1. Promote reuse of ontology
2. Capture ontology errors during use
3. Profile use of ontology elements during use
4. Compare and map between ontologies
5. Track ontology changes and control versions
6. Other ontology quality and fitness support for maintenance phase (describe capability)

Ontology quality and fitness:
A survey of software support

END